BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

Comparison of Machine Learning Techniques to Predict Cardiovascular Disease

Authors
Mir Mohammad Jaber
Tahmid Imam Raad
Tasfia Tasneem

Supervisor
Dr. Amitabha Chakrabarty
Associate Professor
Department of CSE

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of
B.Sc. Engineering in CSE

Department of Computer Science and Engineering
BRAC University, Dhaka - 1212, Bangladesh

December 2018

We would like to dedicate this thesis to our parents.. . .

Declaration

It is hereby declared that this thesis/project report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Authors:

Mir Mohammad Jaber
Student ID: 15101091

Tahmid Imam Raad
Student ID: 17301219

Tasfia Tasneem
Student ID: 15301083

Supervisor:

Dr. Amitabha Chakrabarty
Associate Professor, Department of Computer Science and Engineering
BRAC University

December 2018

Abstract

The purpose of this thesis is to examine and compare the accuracy of different data mining classication systems through different machine learning techniques to predict cardiovascular disease. This comparison shows the different accuracy rates of different techniques and reasons behind their variations. The Cleveland dataset for heart diseases has been used in this study which contains 303 instances. The data has been divided into two sections named as training and testing datasets. The 10- fold Cross Validation has been used here in order to work with the expanded dataset. The k-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, Gaussian Naive Bayes, Logistic Regression and Deep Belief Network machine learning techniques have been investigated in this research. Besides, ensemble learning method voting classifier has been applied on the data set. By the end of the implementation part, we have found Gaussian Naive Bayes is giving the maximum accuracy in our dataset and deep belief network is performing very poor. The reasons of variations of these different techniques by analyzing their characteristics and behavior with respect to the dataset has been understood by the study conducted for this thesis.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

CV	Cross Validation
DBN	Deep Belief Networks
DT	Decision Tree
GNB	Gaussian Naive Bayes
K-NN	K-Nearest Neighbor
LR	Logistic Regression
RF	Random Forest
SVC	Support Vector Classification
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Motivations

According to World Health Organization (WHO) 17.9 million people die each year due to heart diseases or in common term, heart diseases like heart attacks.[1]. In the last two decades heart disease has been a leading cause of death. The stereotype thinking states that heart disease may only be prominent among the elderly people. However, recent study shows that probability of having heart disease is not confined to age only [2]. Many different factors trigger this disease in the young people also. Besides, doctors depend on certain factors only like age, cholesterol level and blood pressure. Unfortunately, that is often insufficient. Also, information provided by the patients may include redundant and interrelated symptoms and signs especially when the patients suffer from more than one type of disease of the same category. Thus physicians may not able to diagnose it correctly [3]. With this concern in the recent times computer technology and machine learning techniques are being used to develop software to assist doctors in making decision of heart disease in the preliminary stage. Early stage detection of the disease and predicting the probability of a person to be at risk of heart disease can reduce the death rate.[4] Moreover a major challenge faced by health care organizations, is the provision of quality services at affordable costs [5]. So Data mining with intelligent algorithms can be used to cope up with these problems of prediction in medical dataset which includes multiple inputs and expensive diagnosis process. These machine learning techniques enable a better way which leads to correct and low-cost diagnosis.

1.2 Methods

In this report the comparison of different machine learning techniques like k-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest,

Gaussian Naive Bayes, Logistic Regression and Deep Belief Network are implemented to predict heart disease. Different algorithms show different accuracy levels. This comparison shows the different accuracy rates of different techniques and reasons behind their variations. Thus, by studying each algorithm the in-depth analysis and reasons for difference under the same dataset is the main aim of this thesis.

1.3 Objectives

The main objective of this study is to bring out the different results obtained by applying different machine learning algorithms on the dataset. The other purposes of this study are -

1. Obtain a clear idea of the previously mentioned machine learning technique algorithms and see how each of them perform. Analyze the results by understanding each algorithm.
2. Compare the results and find out the reasons behind different behavior of each technique with respect to the same dataset.
3. Perform ensembling to see if it is possible to bring any change in the results.

1.4 Thesis Outline

Remaining part of this thesis report has been organized as follows:

- Chapter 2 briefly reviews some of the machine learning algorithms and related concepts from the literature.
- Chapter 3 describes the heart disease dataset which was used in our thesis.
- Chapter 4 discusses about the result that we got during our experiments and analyzes the result.
- In chapter 5 we conclude the report with a summary of our work and our future plan.

Chapter 2

Literature review

In this chapter we discussed about various machine learning classifiers and previous work on the heart disease.

2.1 Algorithms

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). Classification belongs to the category of supervised learning where the targets also provided with the input data. Next we are going to discuss about some of the supervised classification algorithms.

2.1.1 K-Nearest Neighbor

In 1951, Hodges et al. introduced a non-parametric technique for pattern classification which is popularly known the K-Nearest Neighbour rule [6]. k-nearest-neighbor is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. Based on feature similarity this algorithm, looks at one point on a grid, tries to determine if a point is in group A or B, by looking at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If the majority of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa. The k-nearest-neighbor is an example of a "lazy learner" algorithm because it does not generate a model of the data set beforehand. The only calculations it makes are when it is asked to poll the data point's neighbors. This makes K-NN very easy to implement for data mining. K-NN falls in the supervised learning family of algorithms. Suppose we take x to represent a feature like predator or attribute and y to represent a target

like class or label. The goal here is to learn a function $h : X \rightarrow Y$ so that given an unseen observation x, $h(x)$ can confidently predict the corresponding output y. The k-NN classifier is Non-parametric and Instance-based. The non-parametric feature states that it does not make clear assumptions about the functional form of h, avoiding the dangers of mis-modeling the underlying distribution of the data. And by instance based it means that the algorithm does not precisely learn a model. Instead, it chooses to memorize the training instances. The following is the pseudo code of K-NN algorithm -

K-Nearest Neighbor:

```
Classify(X,Y,x) //X: training data, Y: class labels of X
                //x: unknown sample
for i to m do //m = size of X
    Compute distance d(X,x)
end for

Compute set I containing indices for k smallest distances d(X_i,x)

return majority label for {Y_i where i ∈ I}
```

K-NN can be used for the below type of data

1. Labelled data
2. Small data set
3. Less noisy data
4. Less complex data

2.1.2 Decision Tree

The purpose of decision tree is to divide the data set into smaller data sets based on the descriptive features until a small enough set that contains data points that fall under one label has been obtained. It breaks down a dataset into smallest subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. The decision on which feature to split on is made based on resultant entropy reduction or information gain from the split. The term entropy refers to the the measure of uncertainty of a class in a

subset. In other words entropy is used to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one. The information gain is based on the decrease in entropy after a dataset is split on an attribute. So constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches). Decision trees can handle both categorical and numerical data.

id3(examples, attributes)

```
   //examples are the training examples.
   //attributes is a list
   node = DecisionTreeNode(examples)
   # handle target attributes with arbitrary labels
   dictionary = summarizeExamples(examples, targetAttribute)
   for key in dictionary:
       if dictionary[key] == total number of examples
          node.label = key
          return node
   # test for number of examples to avoid overfitting
   if attributes is empty or number of examples
        < minimum allowed per branch:
      node.label = most common value in examples
      return node
   bestA = the attribute with the most information gain
   node.decision = bestA
   for each possible value v of bestA:
      subset = the subset of examples that have value v for bestA
      if subset is not empty:
         node.addBranch(id3(subset, targetAttribute,
         attributes-bestA))
   return node
```

Information Gain Pseudocode

```
infoGain(examples, attribute, entropyOfSet)
   gain = entropyOfSet
   for value in attributeValues(examples, attribute):
       sub = subset(examples, attribute, value)
       gain -= (number in sub)/(total number of examples) *
       entropy(sub)
return gain
```

Entropy Pseudocode

```
entropy(examples)
   result = 0
   # handle target attributes with arbitrary labels
   dictionary = summarizeExamples(examples, targetAttribute)
   for key in dictionary:
       proportion = dictionary[key]/total number of examples
       result -= proportion * log2(proportion)
  return result
```

2.1.3 Support Vector Machine

the Support Vector Machine algorithm was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 [7].Support Vector Machine is an algorithm which is used in extreme cases. It is a frontier which best segregates the two classes. It is preferred by many people as it produces significant accuracy with less computation. The main objective of this algorithm is to basically find a hyperlane in a N dimensional space where N is the number of features that explicitly separates the data points. There can be many possible hyperlanes that can be chosen for classifying two data points. But the best choice will be the hyperplane which leaves the maximum margin from both the classes. It means the hyperlane which must be in the same distance from the closest element of the two classes and also the distance from the closest element of two classes to the hyperplane is the largest we need to choose that hyperplane.[8]

```
candidateSV = { closest pair from opposite classes }
while there are violating points do
    Find a violator
    candidateSV = candidateSV S violator
    if any α p < 0 due to addition of c to S then
    candidateSV = candidateSV \ p
    repeat till all such points are pruned
    end if
end while
```

2.1.4 Random Forest

The first algorithm for random decision forests was created by Tin Kam Ho[9] using the random subspace method. Random Forest is one of the most popular and powerful algorithm because-

- It’s algorithm and process is very simple
- can be used for both classification and regression task

Random Forest builds multiple decision tree and merges them together to get a more accurate and stable prediction. Most of the time it is trained with bagging method. Bagging method is the combination of all the learning models which increases the overall result and create better accuracy.[10]

Moreover, Random Forest uses almost the same hyper-parameters which are used in decision tree or a bagging classifier. But in this case we don’t need to combine a decision tree with a bagging classifier and we can only use the classifier class of random forest.[10]

While growing the trees, Random Forest adds additional randomness to the model. It searches for the best feature among a random subset of features instead of most important features while splitting the node. For this reason, it generates an wide diversity that generally results in a better and accurate model. So in this algorithm, we consider only a random subset of the features for splitting a node.We can make trees more random by additionally using random thresholds for each feature rather than searching for the best possible threshold.[10][11]

Feature Importance-

- Easily we can measure the important of each features on the prediction making process by looking at how much tree nodes, which uses that features reduce impurity of the forest.
- For each feature it calculates the score after training and scales the results so that the sum of all importance is equal to 1.
- By looking at the feature importance we can choose which feature i should take or which one i should ignore because they do not contribute enough or may be they have no contribution to the prediction process. The more features you have, the more likely our model will suffer from fitting.

```
Precondition: A training set S := (x_1, y_1),..., (x_n,y_n),
features F, and number of trees in forest B.
function RandomForest(S , F)
    H ← ∅
    for i ∈ 1,..., B do
        S (i) ← A bootstrap sample from S
        H ← H ∪ {h_i}
    end for
 return H
```

```
function RandomizedTreeLearn(S , F)
   At each node:
       f ← very small subset of F
       Split on best feature in f
return The learned tree
```

Precondition: A training set S := (x1, y1), . . . ,(xn, yn), features F, and number of trees in forest B.

2.1.5 Gaussian Naive Bayes

In machine learning, Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem This Classification is named after Thomas Bayes (1702-1761), who proposed the Bayes Theorem [12]. Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. It makes two fundamental assumption about the dataset that all the features are independent and equal.This classifier uses the Bayes theorem to classify. Equation for this algorithm is -

$$P(X|c_i) = \prod_{k=1}^{n} P(x_k|c_i) = P(x_1|c_i) \cdot P(x_2|c_i) \cdot P(x_2|c_i) \cdot \cdots \cdot P(x_n|c_n) \tag{2.1}$$

Pseudo code for this classification algorithm -

Learning Phase:

For each class value C_i
Computer P(C_i)
For each attribute X_i
    //Compute Distribution D_{ij}
    if attribute X_j is discrete
        D_{ij} = Categorical Distribution for C_i and X_j
    else
        D_{ij} Normal Distribution for C_i and X_j

Testing Phase:

Given unknown $X^{'}$ = [$x_1^{'}$, $x_2^{'}$,...,$x_n^{'}$]
estimate class = $argmax_{C_i}\ P(C_i)\ \cdot\ \prod_j\ D_{ij}$

2.1.6 Logistic Regression

For examining a dataset Logistic regression is a static method in which one or more independent variables will be there and they will determine the outcome of the dataset. Only two possible outcomes are possible and that outcome is measured with dichotomous variable. In Linear Regression algorithm the values which are larger than 1 and less than 0 are ignored fully [13] and that is why we use Logistic regression to solve this problem.[14]

In logistic regression the dependent variable is binary or dichotomous. It contains data coded as 1 which means True or Success or Pregnant etc or it may contain 0 which means False or Unsuccess. It generates the co-efficient of a formula to predict a logit transformation of the probability of presence of characteristic of interest.[14]

$$Logit(P) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \cdots + b_k \cdot x_k$$

Where P is the probability of presence of the characteristic of interest.

$$Logit(P) = \ln \frac{p}{(1-p)}$$

(Probability of presence of characteristics/Probability of absence of characteristics)

To estimate the accuracy and generate the output, Logistic Regression chooses those parameters that maximize likelihood of observing the sample values rather than choosing parameters that minimize the sum of squarred values.[14]

The main goal of this algorithm is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest which is independent variables=response or outcome variable and set of independent variables (predictor or explainatory).[15]

2.1.7 Deep Belief Networks

A deep belief network (DBN) [16] is a sophisticated type of generative neural network that uses an unsupervised machine learning model to produce results. This type of network illustrates some of the work that has been done recently in using relatively unlabeled data to build unsupervised models. Deep belief network is considered a set of restricted Boltzmann machines (RBMs) stacked on top of one another. In general, deep belief networks are composed of various smaller unsupervised neural networks. One of the common features of a deep belief network is that although layers have connections between them, the network does not include connections between units in a single layer.

Geoff Hinton, one of the pioneers of this process, characterizes stacked RBMs

as providing a system that can be trained in a “greedy” manner and describes deep belief networks as models “that extract a deep hierarchical representation of training data.” [16].

In general, this type of unsupervised machine learning model shows how engineers can pursue less structured, more rugged systems where there is not as much data labeling and the technology has to assemble results based on random inputs and iterative processes.

2.2 Related Work

Given the current statistics showing high number of death due to heart diseses [17] numerous people have developed prediction model to predict heart disease. In the past few years many studies have been conducted based on different classication algorithms applied to the Cleveland heart disease dataset freely available at an online data mining repository of the UCI. This dataset has served great purpose to many researchers who conducted research in the field of prediction of cardiovascular diseases.

Echouffo-Tcheugui et al [18] shows a comparative study and a summary of performances of the various statistical approaches which presents 15 different hypertension prediction risk models from 11 studies which report on the development, validation, and impact analysis of hypertension risk prediction models. They have used some common and uncommon variable models such as age, sex, BMI, diabetes status, blood pressure, smoking, family history etc. However, None of these models apply a neural network approach but neural networking is very much efficient in many data model. So they have discussed some of the ANN approaches.

Poli et al [19]. has done a nice work in this field which uses an ANN called Hypernet to diagnose and treat hypertensive patients. The network takes anamnestic data as well as a time series of blood pressure monitoring data as inputs and for outputs they check the the quantity of antihypertensive drugs to every patient and if there is no output it means patient is not hypertensive.

Samant and Rao [20] developed a Levenberg-Marquardt backpropagation neural network in Matlab which takes 13 inputs such as blood pressure, serum proteins, albumin, hematocrit, cholesterol, triglycerides, and hemorheological parameters and gives an output. The authors also evaluated differences in performance based on the number of hidden nodes and layers to determine optimal performance. They also showed that a deep network with 20 input nodes in the first hidden layer and 5 nodes in the second hidden layer increases the efficiency and gives the best result and gives 92.85accuracy.

Ture et al [21]. compared the performances of three decision tree models, four statistical algorithmic models and two ANN models and all of these predict

the risk of hypertension disease. In this process they used the variables such as age, sex, family history, smoking habits, lipoproteins, triglycerides, uric acid, cholesterol, and BMI. For the sensitive and specific analysis of the models it was proved that the variables that they used are very good predictor variables for diagnosing hypertension and ANN models are the best that have the incremental learning capability to complement the existing statistical models.

Srivastava et al [22]. implemented a fuzzy soft computing approach to differentiate five different grades of hypertension and they have come up with the levels like- very low, low, moderate, high, very high. They used age, systolic blood pressure (SBP), diastolic blood pressure (DBP), BMI, heart rate, low density lipoprotein (LDL), high density lipoprotein (HDL), triglyceride, smoking, and exercise as input variables. However, The approach is unable to learn as ANNs do and depends on the definition of the fuzzy utilities.

Sumathi and Santhakumaran [23] used feed-forward backpropagation network and they took eight input variables, four hidden variables, and two output variables achieve results comparable to physicians. However, they did not conclude any Accuracy rate at the end.

Palaniapan et al [24] have developed a prototype Intelligent Heart Disease Prediction System, using techniques like Naïve Bayes, Decision Trees and Neural Network. Various attributes like age, blood pressure, blood sugar. This system is able to predict the possibility of patient getting heart disease.

In our thesis we intended to highlight a comparison of almost all the aforementioned techniques that have been utilized in foregoing studies and their combinations, so that we can conclude on which of the algorithms have shown the highest accuracy.

Chapter 3

Dataset

In this chapter we discussed about the dataset that we used for our thesis. The description of the attributes and about the dataset itself.

3.1 Dataset Description

There are 4 databases concerning heart disease diagnosis in the UCI Machine Lerarning Heart Disease dataset folder. All attributes are numeric valued. The data was gathered from the four following locations:

1. Cleveland Clinic Foundation (cleveland.data)
2. Hungarian Institute of Cardiology, Budapest (hungarian.data)
3. V.A. Medical Center, Long Beach, CA
4. University Hospital, Zurich, Switzerland (switzerland.data)

Instance format of each database is the same. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, till this date the Cleveland database is the only one that has been used by ML researchers. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). For this research we worked with the data collected from the Cleveland Clinic Foundation, The number of instances in this dataset is 303. In the figure 3.1 we can see the first 30 instances of our dataset.

Complete attribute description of the dataset is:

1. age: age in years
2. sex:
   (a) 1 = male
   (b) 0 = female
3. cp: chest pain type
   (a) Value 1: typical angina
   (b) Value 2: atypical angina
   (c) Value 3: non-anginal pain
   (d) Value 4: asymptomatic
4. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholestoral in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl)
   (a) 1 = true
   (b) 0 = false
7. restecg: (resting electrocardiographic results)
   (a) Value 0: normal
   (b) Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
   (c) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. thalach: maximum heart rate achieved
9. exang: exercise induced angina
   (a) 1 = yes
   (b) 0 = no
10. oldpeak: ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment
    (a) Value 1: upsloping

   (b) Value 2: flat

   (c) Value 3: downsloping

12. ca: number of major vessels (0-3) colored by flourosopy

13. thal: thalium heart scan

   (a) 3 = normal (no cold spots)

   (b) 6 = fixed defect (cold spots during rest and exercise)

   (c) 7 = reversible defect (when cold spots only appear during exercise)

14. pred_attribute: (the predicted attribute) diagnosis of heart disease (angiographic disease status)

   (a) Value 0: < 50% diameter narrowing

   (b) Value 1: > 50% diameter narrowing (in any major vessel: attributes 59 through 68 are vessels)

3.2 Review all features

It is very important to know the data. A histogram is a quick way to get information about a sample distribution without detailed statistical graphing or analysis. This plot will show us if our data values are centered (normally distributed), skewed to one side or the other, or have more than one 'mode' - localized distribution concentrations. Now, We can inspect the distribution of the target variable (an imbalanced distribution of target variable might harm the performance of some models).

First in the Figure 3.2 we tried to see frequency distribution for all the participants. By all the participants we mean all those who have heart disease and those who haven't.

In the Figure 3.3 we tried to see frequency distribution for all the heart disease patient.

1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slop	ca	thal	pred_attribute
2	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
3	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
4	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
5	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
6	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
7	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
8	62	0	4	140	268	0	2	160	0	3.6	3	2	3	3
9	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
10	63	1	4	130	254	0	2	147	0	1.4	2	1	7	2
11	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1
12	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
13	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
14	56	1	3	130	256	1	2	142	1	0.6	2	1	6	2
15	44	1	2	120	263	0	0	173	0	0	1	0	7	0
16	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0
17	57	1	3	150	168	0	0	174	0	1.6	1	0	3	0
18	48	1	2	110	229	0	0	168	0	1	3	0	7	1
19	54	1	4	140	239	0	0	160	0	1.2	1	0	3	0
20	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0
21	49	1	2	130	266	0	0	171	0	0.6	1	0	3	0
22	64	1	1	110	211	0	2	144	1	1.8	2	0	3	0
23	58	0	1	150	283	1	2	162	0	1	1	0	3	0
24	58	1	2	120	284	0	2	160	0	1.8	2	0	3	1
25	58	1	3	132	224	0	2	173	0	3.2	1	2	7	3
26	60	1	4	130	206	0	2	132	1	2.4	2	2	7	4
27	50	0	3	120	219	0	0	158	0	1.6	2	0	3	0
28	58	0	3	120	340	0	0	172	0	0	1	0	3	0
29	66	0	1	150	226	0	0	114	0	2.6	3	0	3	0
30	43	1	4	150	247	0	0	171	0	1.5	1	0	3	0

Fig. 3.1 First 5 intances of the Cleveland Dataset

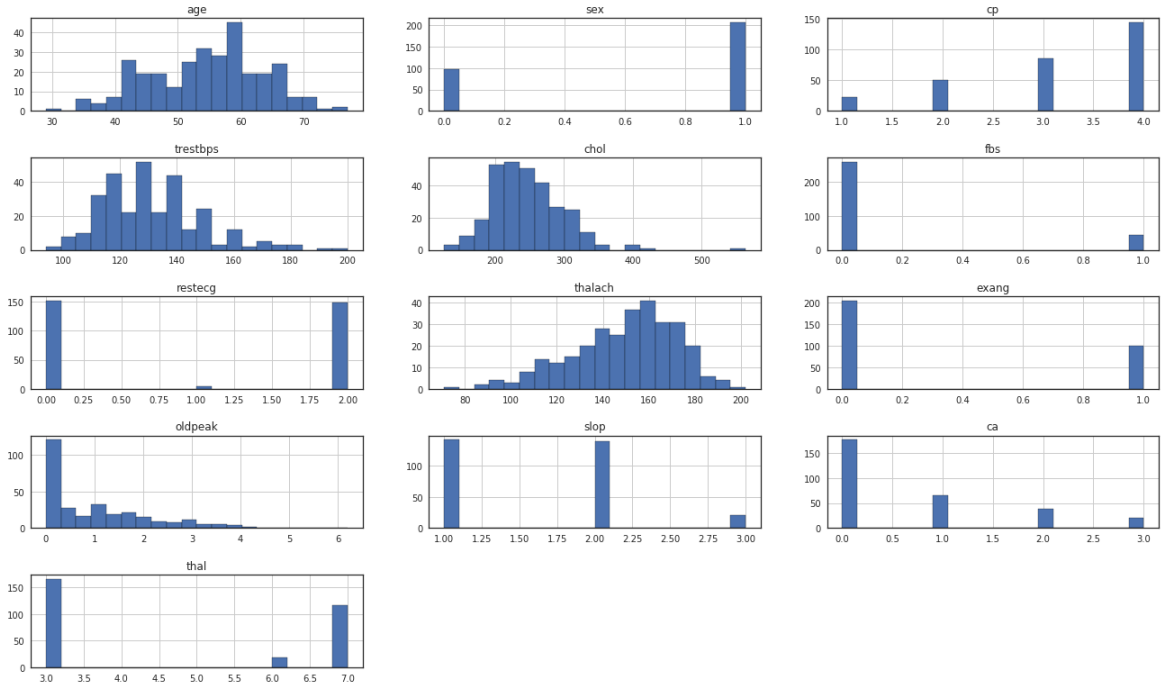

Fig. 3.2 Frequency Distribution of Features for All Participants

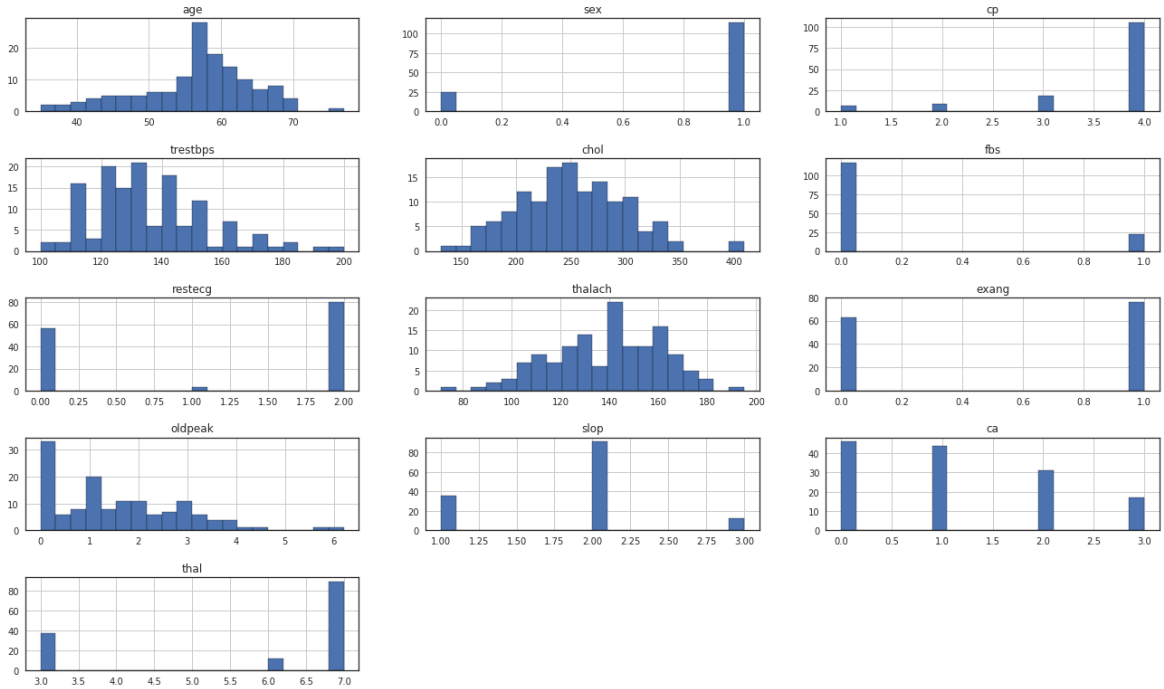

Fig. 3.3 Frequency Distribution of Features for Heart Disease Patients

3.3 Data Preprocessing

It is not quite possible to get well organized data from real world as we want it to be. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. In Real world data are generally -

- Incomplete: lacking attribute values, lacking certain attributes of interest or containing only aggregate data.
- Noisy: containing errors or outliers.
- Inconsistent: containing discrepancies in codes or names.

There were also few missing values in our dataset. In the table 3.1 we showed the number of missing values for each attribute.

Attribute	Number of Missing Value
age	0
sex	0
cp	0
trestbps	0
chol	0
fbs	0
restecg	0
thalach	0
exang	0
oldpeak	0
slop	0
ca	4
thal	2
pred_attribute	0

Table 3.1 Missing Values in dataset

Imputation

In real word dataset it is common to have few missing values. There are few techniques to handle missing values. Two prominent ways to handle Missing Values are described below:

1. This method commonly used to handle the null values. Here, we either delete a particular row if it has a null value for a particular feature and a particular column if it has more than 75% of missing values. This method is

advised only when there are enough samples in the data set. One has to make sure that after we have deleted the data, there is no addition of bias. Removing the data will lead to loss of information which will not give the expected results while predicting the output.

2. This strategy can be applied on a feature which has numeric data. We can calculate the mean, median or mode of the feature and replace it with the missing values. This is an approximation which can add variance to the data set. But the loss of the data can be negated by this method which yields better results compared to removal of rows and columns. Replacing with the above three approximations are a statistical approach of handling the missing values. This method is also called as leaking the data while training. Another way is to approximate it with the deviation of neighbouring values. This works better if the data is linear.

As our dataset is not large enough to delete any instance, We used the second method to deal with the missing values.

Standardization

There can be a lot of deviation in the given dataset. High variance has to be standardized. Standardization, or normalization.
Data standardization is the process of rescaling one or more attributes so that they have a mean value of 0 and a standard deviation of 1.
Standardization assumes that the data has a Gaussian (bell curve) distribution. This does not strictly have to be true, but the technique is more effective if the attribute distribution is Gaussian.

Stratification

When we split the dataset into train and test datasets, the split is completely random. Thus the instances of each class label or outcome in the train or test datasets is random. Thus we may have many instances of class 1 in training data and less instances of class 2 in the training data. So during classification, we may have accurate predictions for class1 but not for class2. Thus we stratify the data, so that we have proportionate data for all the classes in both the training and testing data.

Define training and test samples: The Cleveland data set available from the UCI repository has 303 samples; the training and test data sets were randomly selected with 20% of the original data set corresponding to the test data set. The relative proportions of the classes of interest (disease/no disease) in both sets were checked to be similar.

Error from Imputation

We used the dataset before imputation and after the imputation to train. And tried to do prediction. Mean Absolute Error from imputation: 0.2699453551912568

Chapter 4

Results and Analysis

In our previous chapters we have discussed about different algorithms, previous works in this field and the dataset we used for our experiments. All those were the foundation for this chapter. In this chapter we discussed about results that we found after implementing the algorithms and analyzed them.

4.1 Accuracy of Models with All Features

The results have been obtained by applying different classification and association algorithm. In our frst experiment we used the whole dataset with all features and applied the K-Nearest Neighbors, Support Vector Machine, Decision Tree, Random Forest, Gaussian Naive Bayes, Logistic Regression and Deep Belief Network. Table 4.1 contains accuracy of the different algorithms that we applied on our dataset. Fig 4.1 shows the graphical representation of the table.

Classifier	Accuracy(%)
k-Nearest Neighbors(k=3)	77.0492
k-Nearest Neighbors(k=9)	80.3279
k-Nearest Neighbors(k=15)	86.8852
Support Vector Machine	78.6885
Decision Tree	68.8525
Random Forest	78.6885
Gaussian Naive Bayes	86.8852
Logistic Regression	80.3279
Deep Belief Network	70.4918

Table 4.1 Accuracy of Models with All Features

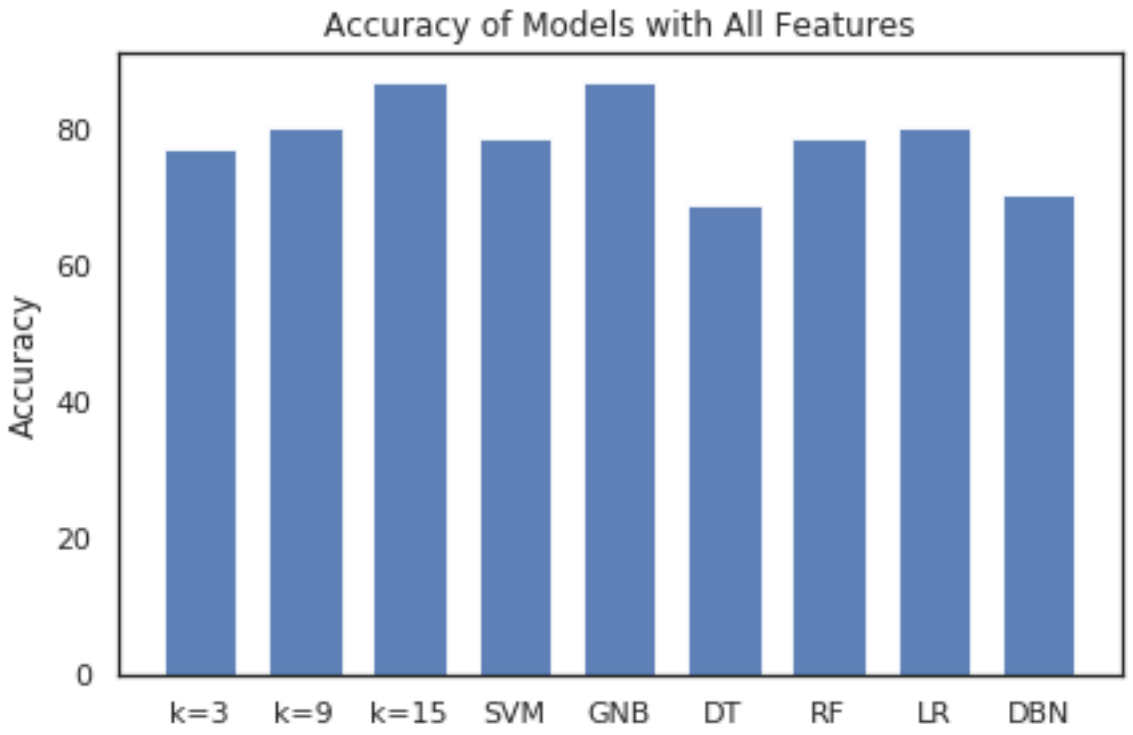

Fig. 4.1 Accuracy of Models with All Features

4.2 Feature Engineering

A lot of features can affect the accuracy of the algorithm. So working with the features is very important. There are few reasons for which some may want to work with some selected features.

1. Choosing less features helps us to train faster.
2. Some features are linearly related to others. This might put a strain on the model.
3. By picking up the most important features, we can use interactions between them as new features. Sometimes this gives surprising improvement.
4. Feature Selection means to select only the important features in-order to improve the accuracy of the algorithm.
5. It reduces training time and reduces overfitting.

Feature Importance

A very basic question that we might ask of a model is What features have the biggest impact on predictions? This concept is called feature importance. In

dataset there may be some attributes which don't effect the prediction that much. In some cases few attributes may decrease the accuracy level of a model. So, it is important to work with the correct attributes. So far we have worked with all the features of the dataset and listed [4.1] the accuracy of different models. Now, we want to see the change of accuracy's of different classifiers after selecting a subset of the attributes. We can see the importance of a feature via Decision Tree [Table 4.2] and Random Forest [Table 4.3].

Attribute	Importance(%)
thal	27.5118
cp	15.6020
oldpeak	11.8239
ca	10.2086
age	9.2777
chol	6.4645
trestbps	6.2695
thalach	5.5372
restecg	2.2596
sex	1.8941
slop	1.2523
fbs	1.0127
exang	0.8861

Table 4.2 Decision Trees Classification - Feature Importance

Attribute	Importance(%)
thal	13.1755
cp	12.0631
thalach	11.4539
ca	11.3193
oldpeak	10.7748
age	8.5289
trestbps	7.9821
chol	7.9433
exang	5.6739
slop	5.3638
sex	3.0252
restecg	1.8785
fbs	0.8178

Table 4.3 Random Forest Classification - Feature Importance

Accuracy of Models with Selected Features

After seeing the feature importance of Table 4.2 and Table 4.3, we selected the below features to see the difference in prediction -

- ca
- thal
- thalach
- cp
- oldpeak
- age
- pred_attribute (Label)

Table 4.4 shows the changes in the accuracy after selecting features. And fig 4.2 shows the graphical representation of changes in the accuracy.

Classifier	New Accuracy(%)	Previous Accuracy(%)	Accuracy Increase(%)
k-Nearest Neighbors(k=3)	80.3279	77.0492	3.2787
k-Nearest Neighbors(k=9)	83.6066	80.3279	3.2787
k-Nearest Neighbors(k=15)	81.9672	86.8852	-4.9180
Support Vector Machine	81.9672	78.6885	3.2787
Decision Tree	72.1311	68.8525	3.2787
Random Forest	80.3279	78.6885	1.6393
Gaussian Naive Bayes	90.1639	86.8852	3.2787
Logistic Regression	83.6066	80.3279	3.2787
Deep Belief Network	80.3278	70.4918	9.8360

Table 4.4 Accuracy of models with selected features

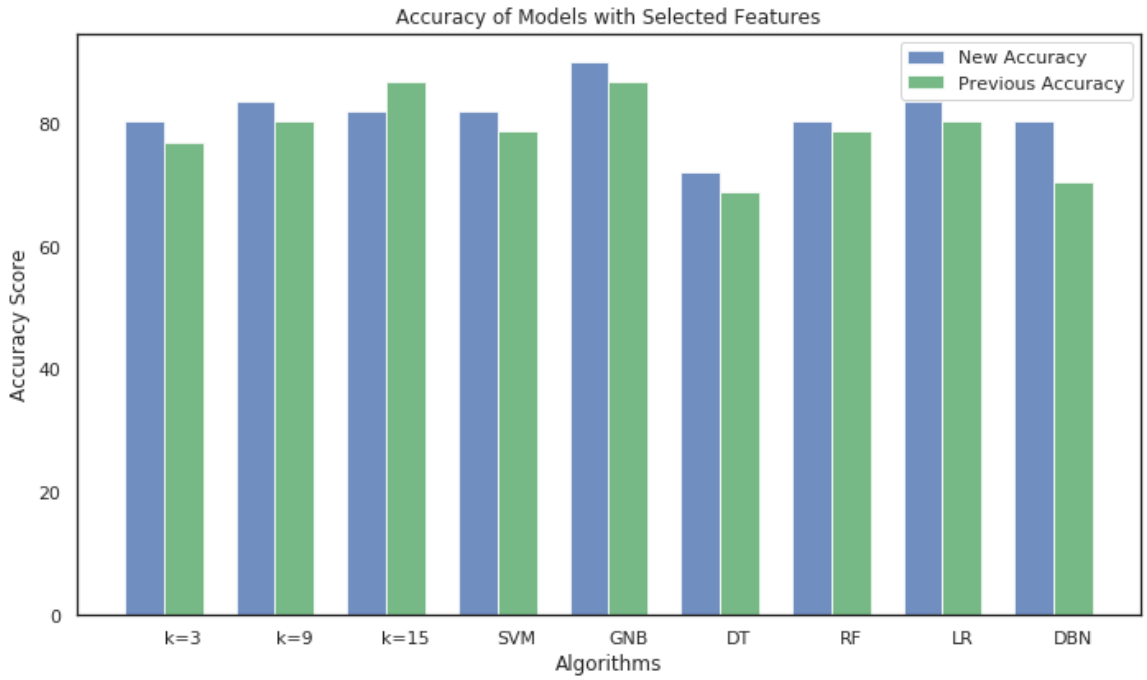

Fig. 4.2 Accuracy of Models with Selected Features

4.3 Cross Validation

Cross validation is an essential step in model training. It tells us whether our model is at high risk of overfitting. In many competitions, public LB scores are not very reliable. Often when we improve the model and get a better local CV score, the LB score becomes worse. It is widely believed that we should trust our CV scores under such situation. Ideally we would want CV scores obtained by different approaches to improve in sync with each other and with the LB score, but this is not always possible.

Usually 5-fold CV is good enough. If we use more folds, the CV score would become more reliable, but the training takes longer to finish as well. However, we shouldn't use too many folds if our training data is limited. Otherwise we would have too few samples in each fold to guarantee statistical significance.

Many times the data is imbalanced, i.e there may be a high number of class1 instances but less number of other class instances. Thus we should train and test our algorithm on each and every instance of the dataset. Then we can take an average of all the noted accuracies over the dataset.

1. The k-Fold Cross Validation works by first dividing the dataset into k-subsets.

2. Let's say we divide the dataset into (k=10) parts. We reserve 1 part for testing and train the algorithm over the other 9 parts.

3. We continue the process by changing the testing part in each iteration and training the algorithm over the other parts. The accuracies and errors are then averaged to get a average accuracy of the algorithm.

4. An algorithm may underfit over a dataset for some training data and sometimes also overfit the data for other training set. Thus with cross-validation, we can achieve a generalised model.

Table - 4.5 shows the standard metrics for 10-fold cross validation technique. And fig -4.3 shows the graphical representation of the change in train and test accuracy scores.

Learning Curves

Learning curves show the relationship between training set size and the chosen evaluation metric (e.g. RMSE, accuracy, etc.) on the training and validation sets. They can be an extremely useful tool when diagnosing model performance, as they can tell whether the model is suffering from bias or variance.

Fig - 4.4 Shows the learning curve of Gaussian Naive Bayes classifier. With the increasing dataset training accuracy score decreases and cross-validation score increases. We can see high bias and low variance in the learning curve. Fig - 4.5 Shows the learning curve of SVC. During the training the accuracy score decreases with the expanding dataset. The model is performing bad for validation sets suggests that the model has high bias. Fig - 4.6 Shows the learning curve of Logistic Regression. We can notice the low variance and the hiagh bias of the model on the dataset.

4.4 Ensemble Learning

Ensemble learning is a machine learning paradigm where multiple learners are trained to solve the same problem. In contrast to ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to use. An ensemble contains a number of learners which are usually called base learners. The generalization ability of an ensemble is usually much stronger than that of base learners. Actually, ensemble learning is appealing because that it is able to boost weak learners which are slightly better than random guess to strong learners which can make very accurate predictions. So, “base learners” are also referred as “weak learners”. It is noteworthy, however, that although most theoretical analyses work on weak learners, base learners used in practice are not necessarily weak since using not-so-weak base learners often results

Table 4.5 Standard Metrics For 10-Fold Cross Validation Technique

	Accuracy Train(%)	**S. Deviation Train**	**Accuracy Test(%)**	**S. Deviation Test**	**ROC (area)**	**AUC (area)**	**Gmean**	**Precision**	**Recall**	**F1 _score**
k-Nearest Neighbors(k=3)	90.358306	0.765752	80.600000	6.645048	0.775275	0.5	0.638500	0.700000	0.807692	0.750000
k-Nearest Neighbors(k=9)	85.583436	0.961447	81.450000	8.792058	0.803846	0.5	0.678005	0.750000	0.807692	0.777778
k-Nearest Neighbors(k=15)	84.619921	1.609458	80.616667	7.329563	0.865934	0.5	0.769565	0.846154	0.846154	0.846154
Support Vector Machine	90.909821	1.036816	82.250000	8.063033	0.789560	0.5	0.657551	0.724138	0.807692	0.763636
Decision Tree	100.000000	0.000000	72.716667	6.489757	0.703846	0.5	0.566304	0.600000	0.807692	0.688525
Random Forest	98.392170	0.855960	77.683333	6.185310	0.789560	0.5	0.657551	0.724138	0.807692	0.763636
Gaussian Naive Bayes	84.618230	1.119189	83.066667	6.496666	0.870879	0.5	0.767821	0.821429	0.884615	0.851852
Logistic Regression	85.216252	0.950626	82.683333	7.738594	0.798901	0.5	0.680435	0.769231	0.769231	0.769231
Deep Belief Network	56.3362	0.011447	56.2167	0.094246	0.774784	0.5	0.638890	0.846154	0.687500	0.758621

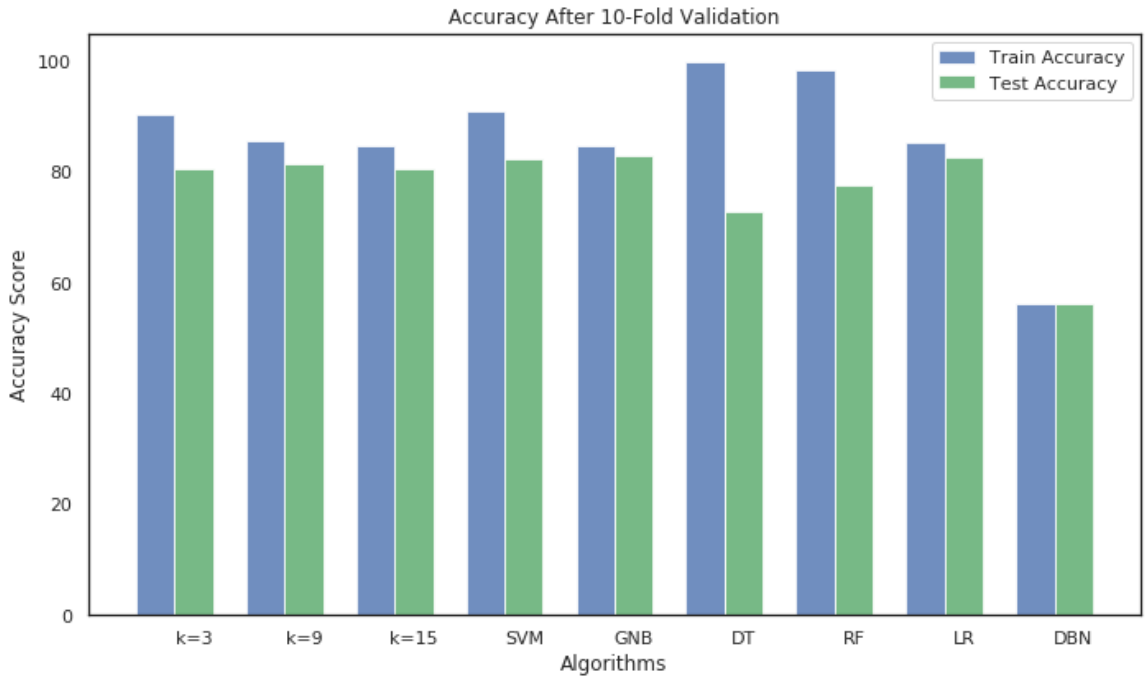

Fig. 4.3 Standard Metrics For 10-Fold Cross Validation Technique

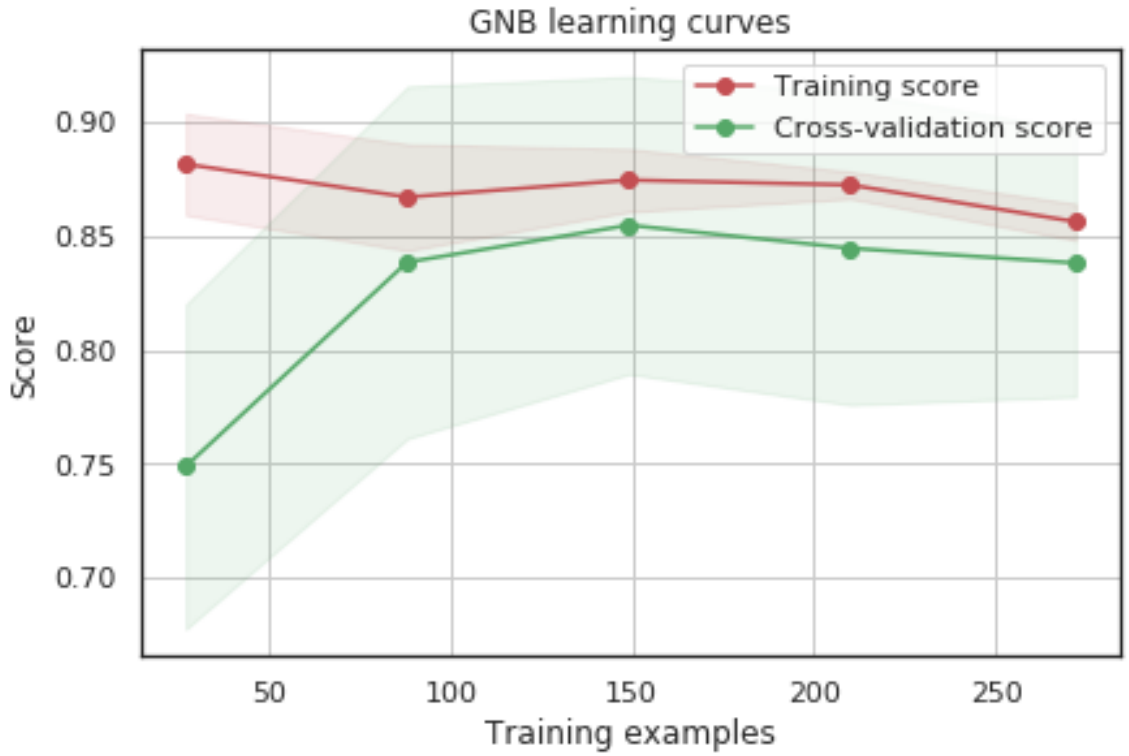

Fig. 4.4 GNB Learning Curve

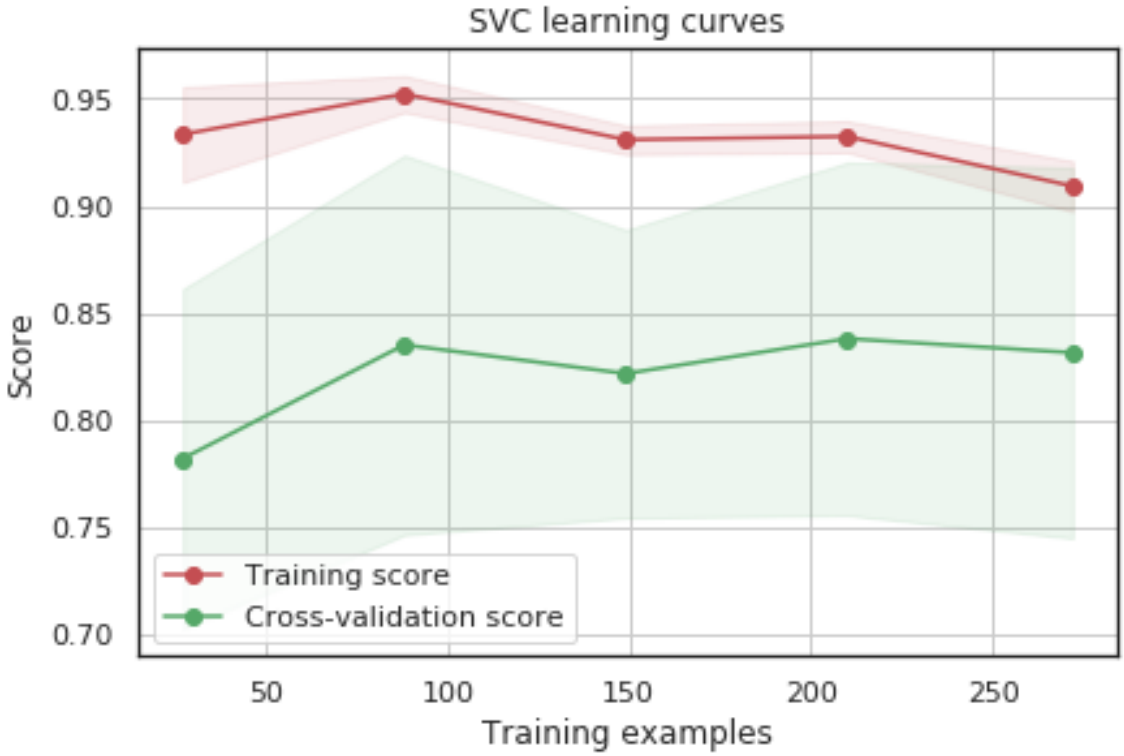

Fig. 4.5 SVC Learning Curve

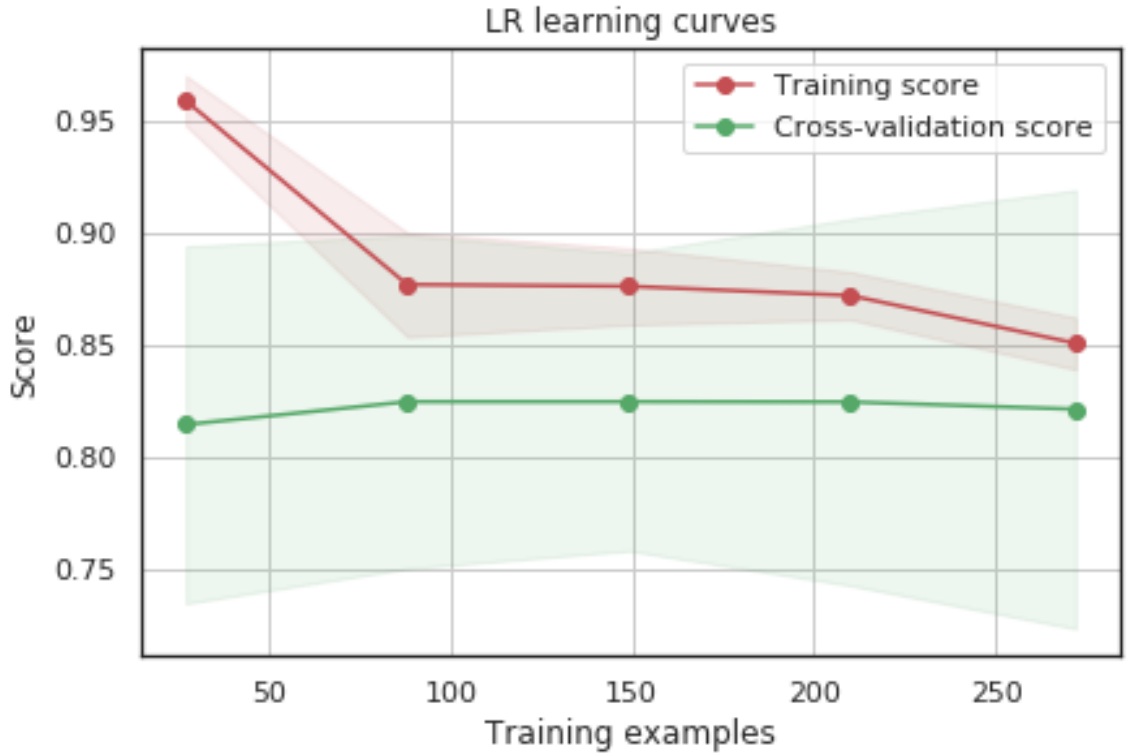

Fig. 4.6 LR Learning Curves

in better performance. Base learners are usually generated from training data by a base learning algorithm which can be decision tree, neural network or other kinds of machine learning algorithms. Most ensemble methods use a single base learning algorithm to produce homogeneous base learners, but there are also some methods which use multiple learning algorithms to produce heterogeneous learners. In the latter case there is no single base learning algorithm and thus, some people prefer calling the learners individual learners or component learners to “base learners”, while the names “individual learners” and “component learners” can also be used for homogeneous base learners.

Voting Classifier

Voting is one of the simplest ways of combining the predictions from multiple machine learning algorithms. It works by first creating two or more standalone models from your training dataset. A Voting Classifier can then be used to wrap the models and average the predictions of the sub-models when asked to make predictions for new data.

A non-weighted voting classifier will be used to stack the top four base models: Guassian Naive Bayes,Support Vector Machine, Random Forest and Logistic Regression. Table 4.6 shows the different accuracies when we used different combination of the models and also used hard and soft voting.

	Accuracy
GNB, LR and SVC (hard voting)	81.9672131147541
GNB, LR and SVC (soft voting)	83.60655737704918
RF, GNB, LR and SVC (soft voting)	85.24590163934426

Table 4.6 Accuracy of Voting Classifiers

Analysis

From all the tables above , different algorithms performed better depending upon the situation whether cross validation and feature selection is used or not. Every algorithm has its intrinsic capacity to out-perform other algorithm depending upon the situation. For example, Random Forest performs much better with a large number of datasets than when data is small. While support vector machine performs better with a smaller number of datasets. In case of decision tree missing values play a important role. Even after imputing it can’t give the result which it can with a perfect dataset. Gaussian naive bayes is the best classifier on this dataset. The reason of it’s pre-assumption that all the attributes are independent. If there was a dependency between the attributes in the dataset it

would have given less accuracy. K-nearest neighbor's accuracy increases with the number of 'k' we pick. It assures the similarity between the given point and the dataset. DBN works better when the dataset has a large number of instances. It helps DBN to understand the underneath patterns in the dataset and to understand it's behavior.

Chapter 5

Conclusion

In this report we have tried to apply several algorithms and compare the accuracy of those algorithms. The main motive of our report was to comparing the accuracy and analysing the reasons behind the variation of different algorithms. We have used Cleveland dataset for heart diseases which contains 303 instances and used 10-fold Cross Validitation to divide the data into two sections which are training and testing datasets. We have considered 13 attributes and implemented seven different algorithms to analyze the accuracy. By the end of the implementation part, we have found Gaussian Naive Bayes is giving the maximum accuracy level in our dataset which is 86.89 percent and Decission Tree is performing the lowest level of accuracy which is 68.85 percent. Probably for other instances and other datasets other algorithm may work in better way but in our case we have found this result. Moreover if we increase the attributes, may be we can found more accurate result but it will take more time to process and the system will be slower than now as it will be little more complex and will be handling more datas. So considering these possible things we took a decission which is better for us to work with.

Future Scope

The dataset that is used in our thesis is very small and old. Moreover no new dataset regarding heart disease has been introduced so far. There is a need of new dataset and we can collect that from various hospitals of Bangladesh. We can also evaluate the efficiency of each individual classifier and also such classifiers in combination, by employing the bagging, boosting and stacking techniques.

References

[1] https://www.who.int/cardiovascular_diseases/en/). *World Health Organization*, Sep 2018. last accessed 24/11/18.

[2] https://www.mayoclinic.org/diseases-conditions/sudden-cardiac-arrest/in-depth/sudden-death/art-20047571.

[3] Chitra Jegan and V. Seenivasagam. Review of heart disease prediction system using data mining and hybrid intelligent techniques. *International Journal Of Soft Computing*, 3(4):2229–6956, 2013.

[4] https://www.alaskaheart.com/the-importance-of-the-early-detection-of-cardiovascular-disease/. 24/11/18.

[5] K. Wankhade M. Gudadhe and S. Dongre. Decision support system for heart disease based on support vector machine and artificial neural network. pages 741–745, 2010.

[6] B. W. Silverman and M. C. Jones. E. fix and j.l. hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951). *International Statistical Review / Revue Internationale de Statistique*, 57(3):233–238, 1989.

[7] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[8] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47. last accessed 24/11/18.

[9] Leena Vig. Comparative analysis of different classifiers for the wisconsin breast cancer dataset. *Open Access Library Journa*, 1(6), 2014. last accessed 1/12/18.

[10] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd. last accessed 24/11/18.

[11] http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/. last accessed 24/11/18.

[12] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 4–15, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[13] https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html. last accessed 24/11/18.

[14] https://www.medcalc.org/manual/logistic_regression.php. 24/11/18.

[15] https://www.statisticssolutions.com/what-is-logistic-regression/. 24/11/18.

[16] Simon Osindero Geoffrey E. Hinton and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[17] http://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[18] Mika Kivimäki Andre P. Kengne Justin B. Echouffo-Tcheugui, G. David Batty. Risk models to predict hypertension: A systematic review. *PLoS ONE*, 8(7):67370, 2013.

[19] Riccardo Livi Giuseppe Coppini Riccardo Poli, Stephano Cagnoni and Guido Valli. A neural network expert system for diagnosing and treating hypertension. *Computer - Special issue on computer-based medical systems*, 24(3):64–71, 1991.

[20] Rahul Samant and Srikantha Rao. Evaluation of artificial neural networks in prediction of essential hypertension. *International Journal of Computer Applications*, 24:0975 – 8887, 2013.

[21] Dr. A. Santhakumaran B. Sumathi. Pre-diagnosis of hypertension using artificial neural network. *Global Journal of Computer Science and Technology*, 11(2), 2011.

[22] Mevlut Ture, Imran Kurt, A. Turhan Kurum, and Kazim Ozdamar. Comparing classification techniques for predicting essential hypertension. *Expert Syst. Appl.*, 29:583–588, 2005.

[23] Anjali Burande Pankaj Srivastava, Amit Srivastava and Amit Khandelwal2. A note on hypertension classification scheme and soft computing decision making system. *ISRN Biomathematics*, 2013:1–11, 2013.

[24] Sellappan Palaniappan and Rafiah Awang. Intelligent heart disease prediction system using data mining techniques. *IJCSNS International Journal of Computer Science and Network Security*, 8(8):343–350, 2008.